Preconditioning Implicit Simulations of Resistive Fusion Plasmas

2006 SIAM Annual Meeting

MS67 – Advances in Computational MHD

Daniel R. Reynolds (UCSD, *drreynolds@ucsd.edu*)

Ravi Samtaney (PPPL, *samtaney@pppl.gov*)

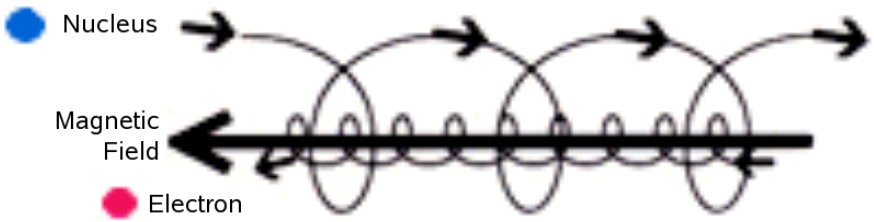
Carol S. Woodward (LLNL, *cswoodward@llnl.gov*)

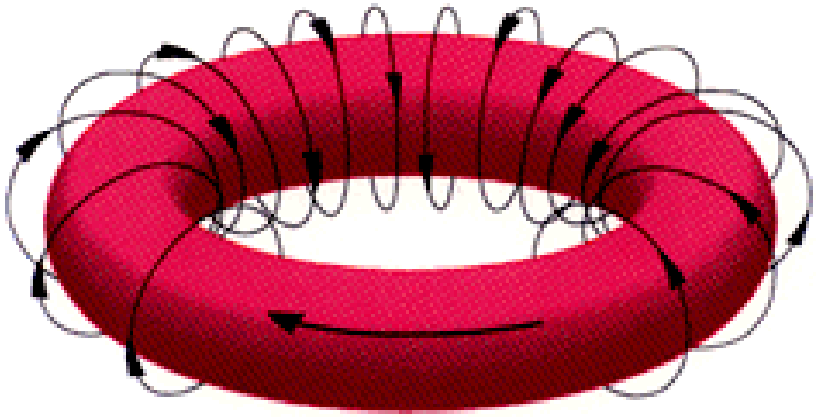
Plasma Confinement

A plasma is a "hot ionized gas" whose motions both generate electromagnetic fields and are influenced by them.

No material walls can contain the plasma at millions of degrees Kelvin. Either the plasma will damage the wall, or the wall will cool the plasma.

Magnetic fusion devides use strong magnetic bottles, in toroidal configurations, to confine the plasma.

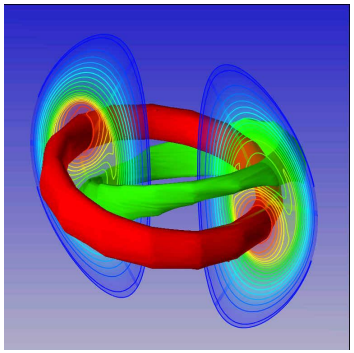




Target Applications

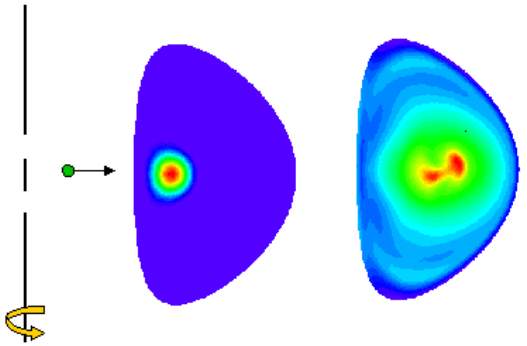
Magnetic Reconnection: [Brin et al., *J. Geophys. Res.*, 2001]

- Replaces hot plasma core with cool plasma
- Interested in investigating macroscopic stability
- Resolution, simulation time scale as $S^{-1/2}, S^{1/2}$
- ITER estimates require $S \approx 10^9$



Pellet Injection Fueling: [Samtaney et al., *Comp. Phys. Comm.*, 2004]

- Shoot hydrogen pellets into plasma at high velocity
- Interested in location of mass deposition
- Pellet motion $\mathcal{O}(10^4)$ slower than fastest waves
- Pellet size $\mathcal{O}(10^4)$ smaller than reactor



Both require *large-scale*, *long-time* simulations, highly anisotropic heat conduction.

Outline

I. MHD Description and Equations

II. Spatio-Temporal Discretization Approach

III. Un-Preconditioned Results

IV. Preconditioning Approach & Results

V. Current Work

Single-Fluid Resistive MHD

Combining conservation laws for a single plasma fluid, Maxwell's equations and a pellet ablation model, we have the system of equations over the domain Ω:

$$\begin{aligned}
\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) &= S_\rho \\
\partial_t (\rho \mathbf{v}) + \nabla \cdot \left(\rho \mathbf{v}\mathbf{v}^T - \mathbf{B}\mathbf{B}^T + \left(p + \tfrac{1}{2}\mathbf{B}\cdot\mathbf{B}\right)\bar{\mathrm{I}} \right) &= \nabla \cdot \bar{\tau} \\
\partial_t e + \nabla \cdot \left((e + p + \tfrac{1}{2}\mathbf{B}\cdot\mathbf{B})\mathbf{v} - \mathbf{B}(\mathbf{B}\cdot\mathbf{v}) \right) &= S_e \\
+ \nabla \cdot (\bar{\tau}\mathbf{v}) + \nabla \cdot \left(\kappa \nabla T + \eta \left(\tfrac{1}{2}\nabla(\mathbf{B}\cdot\mathbf{B}) - \mathbf{B}(\nabla\mathbf{B})^T \right) \right) & \\
\partial_t \mathbf{B} + \nabla \times (\mathbf{v} \times \mathbf{B}) &= \nabla \times (\eta (\nabla \times \mathbf{B})) \\
\partial_t r_p &= S_{r_p}
\end{aligned}$$

where $e = \frac{p}{\gamma - 1} + \rho \frac{\mathbf{v}\cdot\mathbf{v}}{2} + \frac{\mathbf{B}\cdot\mathbf{B}}{2}$, $T = \frac{p}{\rho\, r_{\text{gas}}}$, and $\bar{\tau} = \mu \left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T \right) - \frac{2}{3}\mu(\nabla\cdot\mathbf{v})\bar{\mathrm{I}}$

- $\rho, e \in H^2(\Omega; I\!R^+)$; $\mathbf{v}, \mathbf{B} \in H^2(\Omega; I\!R^3)$; $r_p \in I\!R^+$
- Reconnection: $\Omega \subset I\!R^2$; $S_\rho = S_e = r_p = 0$
- Pellet injection: $\Omega \subset I\!R^3$

Single-Fluid Resistive MHD

We solve these in conservation form for $\mathbf{U} = (\rho,\ \rho\mathbf{v},\ \mathbf{B},\ e,\ r_p)^T$:

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}_h(\mathbf{U}) - \nabla \cdot \mathbf{F}_v(\mathbf{U}) = S,$$

where

$$\mathbf{F}_h(\mathbf{U}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v}^T - \mathbf{B}\mathbf{B}^T + (p + \frac{1}{2}\mathbf{B}\cdot\mathbf{B})\bar{\mathrm{I}} \\ \mathbf{v}\mathbf{B}^T - \mathbf{B}\mathbf{v}^T \\ (e + p + \frac{1}{2}\mathbf{B}\cdot\mathbf{B})\mathbf{v} - \mathbf{B}(\mathbf{B}\cdot\mathbf{v}) \\ 0 \end{pmatrix}, \longleftarrow \text{Hyperbolic flux (Ideal MHD)}$$

$$\mathbf{F}_v(\mathbf{U}) = \begin{pmatrix} 0 \\ \bar{\tau} \\ \eta\left(\nabla\mathbf{B} - (\nabla\mathbf{B})^T\right) \\ \bar{\tau}\mathbf{v} + \kappa\nabla T + \eta\left(\nabla(\frac{1}{2}\mathbf{B}\cdot\mathbf{B}) - \mathbf{B}(\nabla\mathbf{B})^T\right) \\ 0 \end{pmatrix}, \longleftarrow \text{Parabolic flux}$$

$$S = \left(S_\rho,\ 0,\ 0,\ S_e,\ S_{r_p}\right)^T \longleftarrow \text{Pellet ablation sources}$$

Outline

I. MHD Description and Equations

II. Spatio-Temporal Discretization Approach

III. Un-Preconditioned Results

IV. Preconditioning Approach & Results

V. Current Work

Finite Volume Spatial Discretization

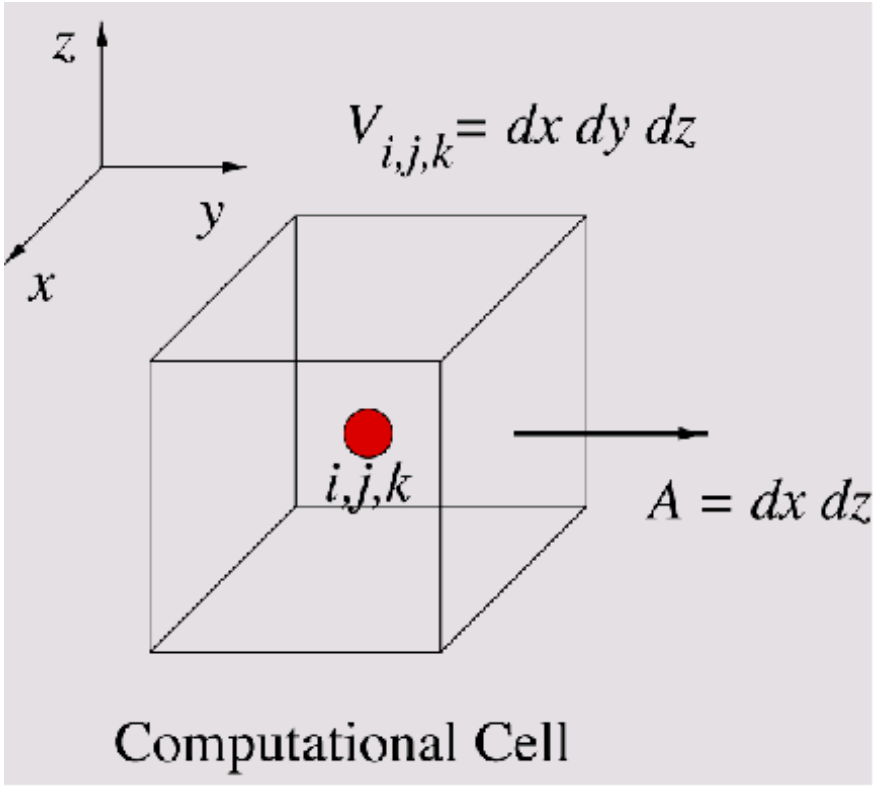
- Divergence form of equations:

$$\partial_t \mathbf{U} = \nabla \cdot \mathbf{F}(\mathbf{U}) + S(\mathbf{U})$$

- Divergence Theorem for volume integral:

$$\partial_t \mathbf{U}_{\text{cell}} = \sum_{\text{faces f}} A_f \mathbf{F}_f \cdot n_f + S_{\text{cell}}$$

- Flux computation at cell faces is the essence of these schemes



Spatial Discretization Details

Interpolation to cell faces can dramatically affect

- Physicality of results ($\nabla \cdot \mathbf{B}$, shock resolution, etc.)
- Numerical stability
- Solution accuracy and computational cost

Our simulations may use any one of:

1. $\mathcal{O}(h^2)$ *Godunov*: Roe and Riemann methods; inherently dissipative; helpful for capturing shocks; non-commutative ($\pi_i \pi_j \neq \pi_j \pi_i$)
2. $\mathcal{O}(h^2)$ *or* $\mathcal{O}(h^4)$ *Centered*: interpolation based on neighbor averages; non-dissipative; increased dispersion error; $\pi_i \pi_j = \pi_j \pi_i$
3. $\mathcal{O}(h^2)$ *Tuned Centered*: same as (2), but with reduced dispersion error
4. $\mathcal{O}(h^2)$ *Zip Average*: similar interpolation properties to (2), possibly improved stability [Chacon, Knoll; Hirt]

Implicit BDF Time Stepping

Method of lines approach for spliting time and space dimensions, time integration based on a high-order (up to $\mathcal{O}(\Delta t^5)$) BDF method:

$$g(\mathbf{U}^n) \equiv \mathbf{U}^n - \Delta t_n \beta_{n,0} f(\mathbf{U}^n) - \sum_{i=1}^{q_n} \left[\alpha_{n,i} \mathbf{U}^{n-i} + \Delta t_n \beta_{n,i} f(\mathbf{U}^{n-i}) \right].$$

- RHS function $f(\mathbf{U}) \equiv \nabla \cdot (\mathbf{F}_v(\mathbf{U}) - \mathbf{F}_h(\mathbf{U})) + S(\mathbf{U})$
- Time-evolved state $\mathbf{U}^n$ solves the nonlinear residual equation $g(\mathbf{U}) = 0$
- q_n determines the method's order of accuracy: at $q_n = \{1, 2\}$ the method is stable for any Δt_n, stability decreases as q_n increases.
- $\alpha_{n,i}$ and $\beta_{n,i}$ are fixed parameters for a given method order q_n
- $\Delta t_n, q_n$ adaptively chosen at each time step to balance solution accuracy, solver convergence, and temporal stability.

[see Gear & Saad, *SIAM J. Sci. Stat. Comput.*, 4, 1983.; Hindmarsh et al., *ACM TOMS*, 31, 2005.]

Inexact Newton-Krylov Solver

Iterate toward $g(\mathbf{U}^n) = 0$ using a sequence of linearized solutions:

1. Given an explicitly-predicted initial guess, $\mathbf{U}_0 \in \text{span}\{\mathbf{U}^{n-i}\}_{i=1}^{q_n}$
2. For each iterate k, until $\|g(\mathbf{U}_{k+1})\| < Ntol$:
   (a) Approximately solve the linear system $J(\mathbf{U}_k)\,\delta\mathbf{U}_k = -g(\mathbf{U}_k)$
   (b) Update the approximate solution: $\mathbf{U}_{k+1} = \mathbf{U}_k + \lambda\,\delta\mathbf{U}_k$

- $\|\cdot\|$ is a 2-norm, weighted by relative magnitudes of expected solution
- Jacobian $J(\mathbf{U})$ provides a linear model of $g(\mathbf{U})$ around $\mathbf{U}$:

$$J(\mathbf{U}) \equiv \frac{\partial}{\partial \mathbf{U}} g(\mathbf{U}) = \frac{\partial}{\partial \mathbf{U}} \left[\mathbf{U} - \overline{\gamma} f(\mathbf{U})\right], \quad \overline{\gamma} \equiv \Delta t_n \beta_{n,0}$$

- Linear systems solved with GMRES iterative solver; requires only products $J(\mathbf{U}^k)\mathbf{V}$, which are approximated so no matrix need be created/stored

$$J(\mathbf{U}_k)\mathbf{V} = \left[g(\mathbf{U}_k + \sigma\mathbf{V}) - g(\mathbf{U}_k)\right]/\sigma + O(\sigma).$$

Properties of this Approach

The BDF-Newton-Krylov approach has a number of attractive properties:

- Newton convergence is independent of refining spatial resolution.
- Retains $\nabla \cdot \mathbf{B} \approx 0$ in time for any nonlinear tolerance $Ntol$ (assuming $\nabla \cdot \mathbf{B}^0 \approx 0$, & commutative spatial differencing).
- If boundary conditions satisfy $\int_{\partial\Omega} \left(\mathbf{F}_{v,i}(\mathbf{U}) - \mathbf{F}_{h,i}(\mathbf{U})\right) \cdot n \, ds = 0$ for each conservation species i, we retain strongly conservative discrete solutions,

$$\int_\Omega \left(\mathbf{U}_i^n - \mathbf{U}_i^{n-1}\right) dx = \int_\Omega \left(S(\mathbf{U}_i^n) - S(\mathbf{U}_i^{n-1})\right) dx, \quad \text{for any } Ntol.$$

The challenges lie within the inner Krylov method:

- For nonsingular $J(\mathbf{U}) \in I\!R^{n\times n}$, GMRES converges in at most n iterations; but each iterate requires storage of an additional $I\!R^n$-vector.
- The convergence rate depends on $\lambda(J(\mathbf{U}))$, and is fastest for tightly-clustered eigenvalues; typically degrades as $\kappa(J)$ increases.

Outline

I. MHD Description and Equations

II. Spatio-Temporal Discretization Approach

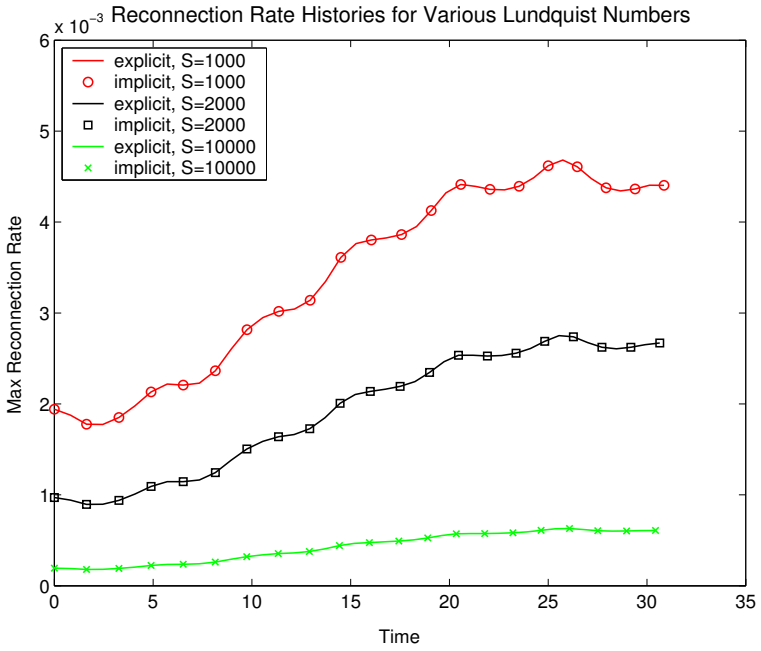
III. Un-Preconditioned Results

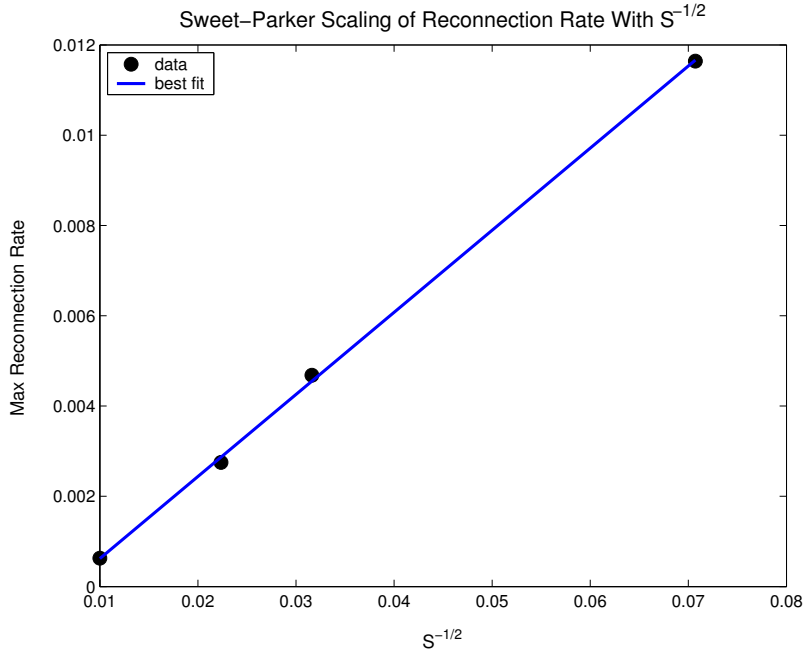
IV. Preconditioning Approach & Results

V. Current Work

Un-Preconditioned Reconnection Results

Implicit method produces identical physical results as explicit, captures theoretical scaling properties.





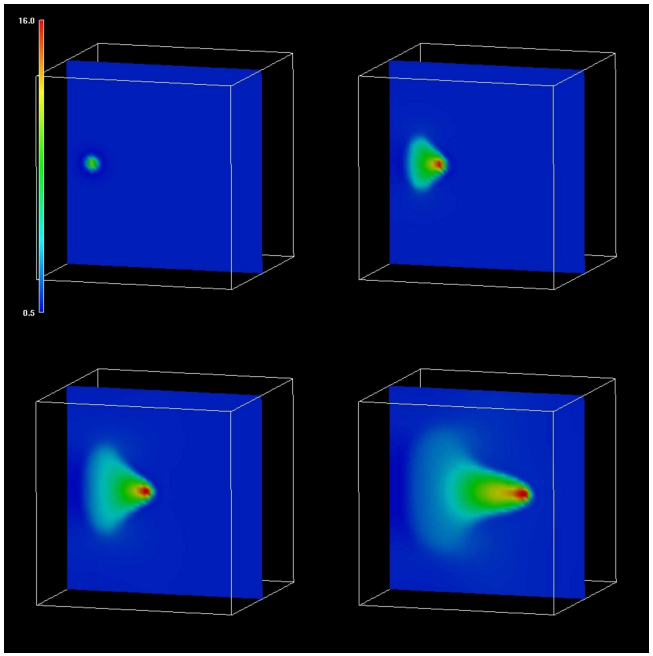
Explicit and implicit CPU times, time steps, step sizes to reach $t = 50$.

mesh size	Exp. CPU Time	Exp Nt	Exp $\overline{\Delta t}$	Imp. CPU Time	Imp Nt	Imp $\overline{\Delta t}$
64×32	75 s.	1636	3.06e-2	47 s.	1144	4.37e-2
128×64	768 s.	3247	1.54e-2	285 s.	1158	4.32e-2
256×128	8214 s.	6493	7.70e-3	1817 s.	1075	4.65e-2
512×256	80348 s.	12985	3.85e-3	14203 s.	1473	3.39e-2

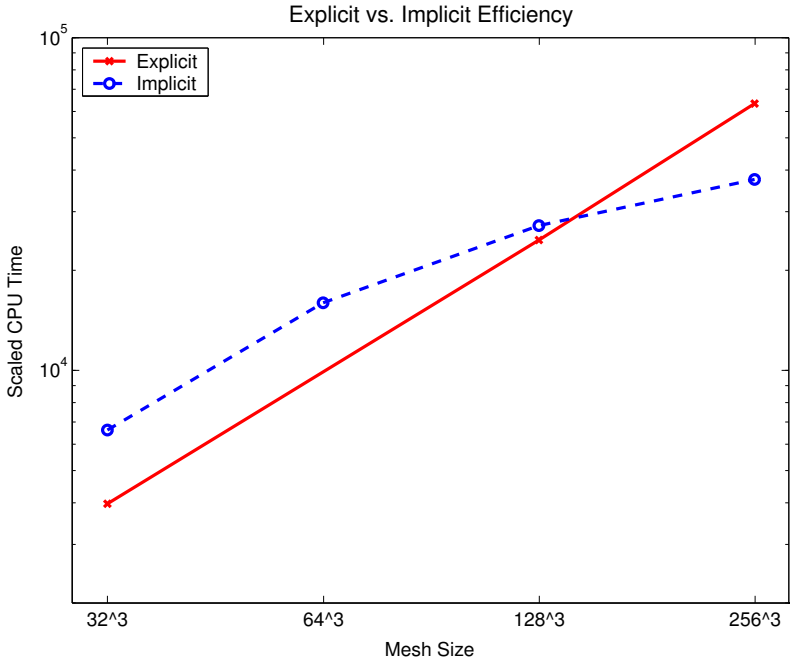
[see R., Samtaney & Woodward, *JCP*, 2006.]

Un-Preconditioned Pellet Injection

Pellet injection/ablation density snapshots;



Explicit, Implicit scaled CPU times.



Explicit and implicit time steps, sizes, CPU times to reach $t = 3$.

mesh size (procs)	Exp. CPU Time	Exp Nt	Exp $\overline{\Delta t}$	Imp. CPU Time	Imp Nt	Imp $\overline{\Delta t}$
32^3 (1)	4198 s.	2844	1.05e-3	7168 s.	6221	4.82e-4
64^3 (8)	9136 s.	4886	6.14e-4	16520 s.	6467	4.64e-4
128^3 (64)	23136 s.	8995	3.34e-4	28598 s.	8979	3.34e-4
256^3 (256)	49507 s.	17619	1.70e-4	40842 s.	9725	3.08e-4

[see R., Samtaney & Woodward, *JCP*, 2006.]

Outline

I. MHD Description and Equations

II. Spatio-Temporal Discretization Approach

III. Un-Preconditioned Results

IV. Preconditioning Approach & Results

V. Current Work

Implicit MHD Preconditioner Acceleration

Instead of solving $J\,\delta\mathbf{U} = -g$, we solve the related system $(JP^{-1})(P\,\delta\mathbf{U}) = -g$.

- P may be any approximation to J that is efficiently solved.
- P approximations does not affect the accuracy of the nonlinear solution, only the Krylov convergence properties.
- P *does* affect the subtler properties of the overall method (e.g. $\nabla \cdot \mathbf{B}$).

Since MHD stiffness results from fast hyperbolic and diffusive effects, we set

$$P^{-1} = P_h^{-1} P_d^{-1} = J(\mathbf{U})^{-1} + \mathcal{O}(\Delta t^2).$$

- Operator-splitting approach widely used as a stand-alone solver, we use it to accelerate convergence of our more stable and accurate approach
- Different problems will require one more than another

P_h: Ideal MHD Preconditioner

P_h treats only the fastest wave effects within the implicit MHD system. Denoting $(\cdot)$ as the location of linear operator action, ideal MHD has Jacobian

$$\begin{aligned}
J_h(\mathbf{U}) &= I + \overline{\gamma}\left[J_x\partial_x(\cdot) + J_y\partial_y(\cdot) + J_z\partial_z(\cdot)\right] \\
&= I + \overline{\gamma}\left[J_xL_x^{-1}L_x\partial_x(\cdot) + J_yL_y^{-1}L_y\partial_y(\cdot) + J_zL_z^{-1}L_z\partial_z(\cdot)\right] \\
&= I + \overline{\gamma}\left[J_xL_x^{-1}\partial_x\left(L_x(\cdot)\right) - J_xL_x^{-1}\partial_x(L_x)(\cdot)\right. \\
&\qquad + J_yL_y^{-1}\partial_y\left(L_y(\cdot)\right) - J_yL_y^{-1}\partial_y(L_y)(\cdot) \\
&\qquad \left. + J_zL_z^{-1}\partial_z\left(L_z(\cdot)\right) - J_zL_z^{-1}\partial_z(L_z)(\cdot)\right]
\end{aligned}$$

We then form an ADI-based $\mathcal{O}(\overline{\gamma}^2)$ preconditioner as

$$\begin{aligned}
P_h = &\left[I + \overline{\gamma}J_xL_x^{-1}\partial_x(L_x(\cdot))\right]\left[I + \overline{\gamma}J_yL_y^{-1}\partial_y(L_y(\cdot))\right]\left[I + \overline{\gamma}J_zL_z^{-1}\partial_z(L_z(\cdot))\right] \\
&\left[I - \overline{\gamma}\left(J_xL_x^{-1}\partial_x\left(L_x\right) + J_yL_y^{-1}\partial_y\left(L_y\right) + J_zL_z^{-1}\partial_z\left(L_z\right)\right)\right] \\
= &\, P_x\, P_y\, P_z\, P_{\text{corr}}.
\end{aligned}$$

P_h: Solving the $P_{x,y,z}$ Systems

- L_i is the spatially-local left eigenvector matrix for J_i, i.e. at a given $x \in \Omega$,

$$L_i(x)J_i(x) = \Lambda_i(x)L_i(x), \quad \Lambda_i = \mathrm{Diag}(\lambda^1, \ldots, \lambda^8)$$

- May be decoupled into 1D wave equations along characteristics:

$$L_i \left[I + \overline{\gamma} J_i L_i^{-1} \partial_i (L_i(\cdot)) \right] \xi = L_i \beta \quad \Leftrightarrow \quad \zeta + \overline{\gamma}\, \Lambda_i\, \partial_i \zeta = \chi,$$

  where $\zeta = L_i \xi$ and $\chi = L_i \beta$.

- We need only solve for the fastest, stiffness-inducing, waves.

- Solved only to low-order ($\mathcal{O}(\Delta x^2)$) since used in preconditioning context $\Rightarrow$ sequence of tridiagonal system solves.

- Parallelized via a divide-and-conquer approach [Arbenz & Gander, 1994]:

  1. Each processor solves local portion of tridiagonal system
  2. Small global correction system constructed and solved on all procs.

P_h: Solving the P_{corr} System

For problems with spatially-dependent $J(\mathbf{U})$, a correction solve is required:

$$\begin{aligned} P_{\text{corr}} &= I - \overline{\gamma}\left[J_x L_x^{-1}\partial_x\left(L_x\right) + J_y L_y^{-1}\partial_y\left(L_y\right) + J_z L_z^{-1}\partial_z\left(L_z\right)\right] \\ &= I - \overline{\gamma}\left[L_x^{-1}\Lambda_x\partial_x\left(L_x\right) + L_y^{-1}\Lambda_y\partial_y\left(L_y\right) + L_z^{-1}\Lambda_z\partial_z\left(L_z\right)\right] \end{aligned}$$

- May be constructed from existing eigen-information.
- Since it has no spatial couplings on the unknown, the resulting local block systems may be solved easily:
  - Pre-compute the 8×8 block-matrices P_{corr} at each location.
  - Factorize each block $P_{\text{corr}} = L_c U_c$.
  - Use these decompositions for fast solves at each Krylov iteration.

P_d: Diffusive MHD Preconditioner

P_d solves the remaining diffusive effects within the implicit system,

$$\partial_t \mathbf{U} - \nabla \cdot \mathbf{F}_v = 0.$$

We set P_d to be the Jacobian of this operator,

$$P_d = J_v(\mathbf{U}) = I - \overline{\gamma} \frac{\partial}{\partial \mathbf{U}} (\nabla \cdot \mathbf{F}_v)$$

$$= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I - \overline{\gamma} D_{\rho \mathbf{v}} & 0 & 0 \\ 0 & 0 & I - \overline{\gamma} D_{\mathbf{B}} & 0 \\ -\overline{\gamma} L_{\rho} & -\overline{\gamma} L_{\rho \mathbf{v}} & -\overline{\gamma} L_{\mathbf{B}} & I - \overline{\gamma} D_e \end{bmatrix}$$

and we then exploit its structure for efficient and accurate solution.

P_d: Implementation Details

To solve $P_d\, y = b$ for $y = [y_\rho, y_{\rho\mathbf{v}}, y_\mathbf{B}, y_e]^T$:

1. Update $y_\rho = b_\rho$
2. Solve $(I - \overline{\gamma} D_{\rho\mathbf{v}})\, y_{\rho\mathbf{v}} = b_{\rho\mathbf{v}}$ for $y_{\rho\mathbf{v}}$
3. Solve $(I - \overline{\gamma} D_\mathbf{B})\, y_\mathbf{B} = b_\mathbf{B}$ for $y_\mathbf{B}$
4. Update $\tilde{b}_e = b_e + \overline{\gamma}\,(L_\rho\, y_\rho + L_{\rho\mathbf{v}}\, y_{\rho\mathbf{v}} + L_\mathbf{B}\, y_\mathbf{B})$
5. Solve $(I - \overline{\gamma} D_e)\, y_e = \tilde{b}_e$ for y_e.

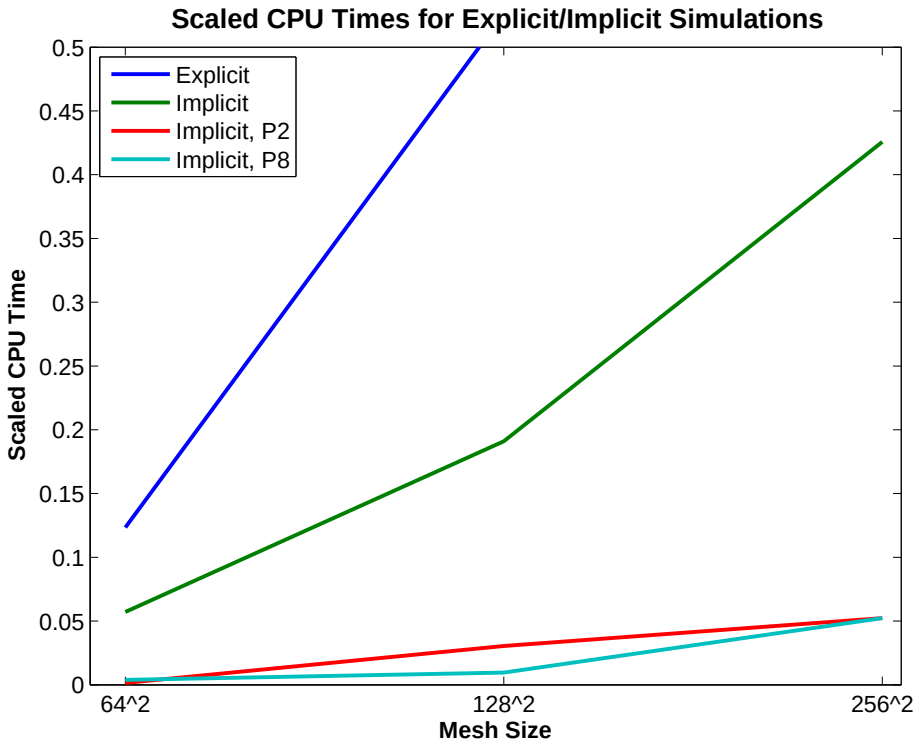
- Due to their diffusive nature, steps 2, 3 and 5 are solved using a system-based geometric multigrid solver [HYPRE, SysPFMG].
- Step 4 may be approximated through one finite-difference, instead of constructing and multiplying by the individual sub-matrices:

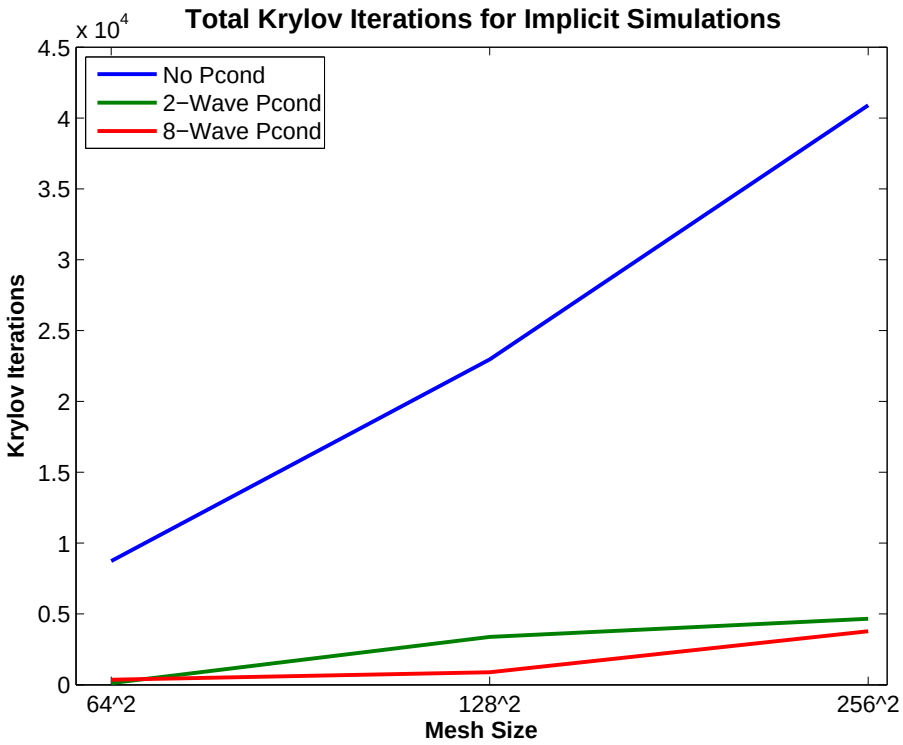
$$L_\rho\, y_\rho + L_{\rho\mathbf{v}}\, y_{\rho\mathbf{v}} + L_\mathbf{B}\, y_\mathbf{B} = \tfrac{1}{\sigma}\left[\nabla \cdot \mathbf{F}_v(U + \sigma W) - \nabla \cdot \mathbf{F}_v(\mathbf{U})\right]_e + O(\sigma),$$

where $W = [y_\rho, y_{\rho\mathbf{v}}, y_\mathbf{B}, 0]^T$.

P_h Results – Ideal MHD Example

- Ideal MHD linear wave propagation test problem to $t = 50$
- Adaptive high-order BDF Implicit (rtol= 10^{-7}, lintol= 10^{-2}); RK4 Explicit
- 2nd-order spatial Centered-Differences



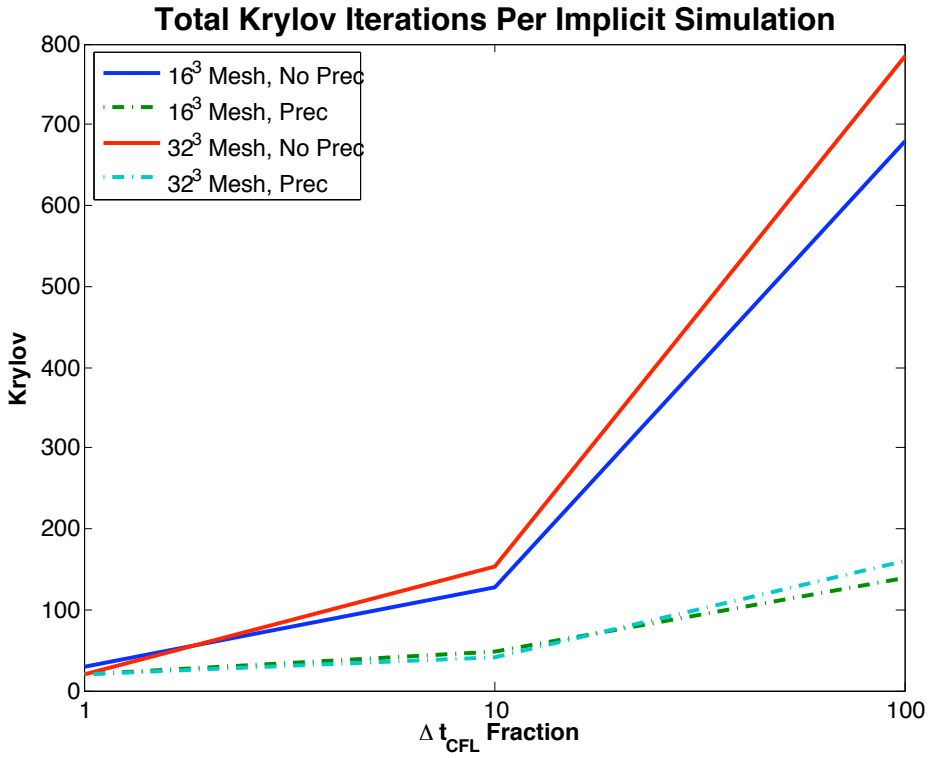


Scaled CPU times computed as (Total CPU Time)/(Total Spatial Cells); horizontal line $\Rightarrow$ perfect weak scaling

Initial P_d Results – Diffusive Example

- Diffusion-dominated resistive MHD pellet-injection test problem ($\eta = 1$)
- 10 Fixed steps, Crank-Nicholson Implicit (Ntol= 10^{-7}, steptol= 10^{-9})
- 2nd-order spatial Centered-Differences

Mesh	Prec	CFL Frac	Newton	Krylov
16^3	N	1	21	30
16^3	Y	1	20	20
16^3	N	10	30	128
16^3	Y	10	40	49
16^3	N	100	34	680
16^3	Y	100	45	140
32^3	N	1	20	20
32^3	Y	1	20	20
32^3	N	10	30	153
32^3	Y	10	40	40
32^3	N	100	35	785
32^3	Y	100	51	160



Outline

I. MHD Description and Equations

II. Spatio-Temporal Discretization Approach

III. Un-Preconditioned Results

IV. Preconditioning Approach & Results

V. Current Work

Current Related Research

- Examine P_h effectiveness on Kelvin-Helmholtz and Tilt-Mode instabilities
- Finish parallel implementation of P_d preconditioner
- Investigate coupled preconditioning approach on pellet-injection and reconnection test problems
- Investigate $O(N)$ divergence-cleaning approach to ameliorate preconditioning-induced $\nabla \cdot \mathbf{B}$ errors [Finn & Chacón, *Physics of Plasmas*, 2005]
- Extend preconditioners to mapped spatial grids, allowing for finite-volume discretization of toroidal fusion devices (Samtaney, MS78)

Conclusions

- For MHD systems dominated by hyperbolic stiffness from fast wave effects, we have developed a preconditioning approach that:
  - uses characteristic info. otherwise used only within upwind methods,
  - allows preconditioning of any combination of MHD waves,
  - is fully parallel, requiring minimal communication per P_h^{-1} solve.
- For MHD systems dominated by diffusive stiffness, we have a multigrid-based preconditioning approach that:
  - uses scalable solver technology for diffusive problems,
  - is easily extensible to highly-anisotropic heat conduction (AMG).
- Most problems exhibit both of these effects, so they may be combined in an operator-split fashion.

Thanks and Acknowledgements

- U.S. Department of Energy, Scientific Discovery through Advanced Computing Program. Specifically:
  - Terascale Optimal PDE Solvers (TOPS),
  - Center for Extended MHD Modeling (CEMM), and
  - Terascale Supernova Initiative

- UCSD Mathematics and Physics Departments

- Lawrence Livermore National Lab:
  - Nonlinear Solvers and Differential Equations Project
  - Scalable Linear Solvers Project

Divergence Cleaning in $O(N)$ Time

After each P^{-1} solve, we wish to post-process $\mathbf{B}$ update to enforce $\nabla \cdot \mathbf{B} \approx 0$. Writing $\mathbf{B} = \nabla \times \mathbf{A}$, with guage $A_x = 0$, construct $\mathbf{A} = (A_x, A_y, A_z)^T$ via:

$$A_y(x,y,z) = \int_{x_{\text{min}}}^{x} B_z(\xi,y,z)d\xi$$

$$A_z(x,y,z) = \int_{y_{\text{min}}}^{y} B_x(x_{\text{min}},\eta,z)d\eta - \int_{x_{\text{min}}}^{x} B_y(\xi,y,z)d\xi$$

1. Compute line integrals to $O(\Delta x^2)$.
2. Interpolate to obtain $A_y(x,y,z)$ and $A_z(x,y,z)$ to $O(\Delta x^3)$.
3. Analytically differentiate $\mathbf{A}$ to obtain $\mathbf{B}$.

Questions:

- What is the accuracy of the resulting $\mathbf{B}$, it's affect on convergence?
- Efficient implementation for non-regular meshes?

[see Finn & Chacón, *Physics of Plasmas*, vol. 12, 2005]

Parallelism of Directional Solves

For each wave, we have a simple 1D tridiagonal system, spread over all processors in a spatial row. We parallelize the solve in two phases:

1. (local) Each processor solves its own portion of the system:

$$\begin{bmatrix} T_1^{-1} & & & \\ & T_2^{-1} & & \\ & & \ddots & \\ & & & T_p^{-1} \end{bmatrix} \begin{bmatrix} T_1 & C_1 & & \\ D_2 & T_2 & \ddots & \\ & \ddots & \ddots & C_{p-1} \\ & & D_p & T_p \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} = \begin{bmatrix} T_1^{-1} & & & \\ & T_2^{-1} & & \\ & & \ddots & \\ & & & T_p^{-1} \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix}$$

$$\Longleftrightarrow \quad \begin{bmatrix} I & E_1 & & \\ F_2 & I & \ddots & \\ & \ddots & \ddots & E_{p-1} \\ & & F_p & I \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{pmatrix}$$

Parallelism of Directional Solves

2. (global) Everythin depends on only a $(2p \times 2p)$ subset of unknowns at processor boundaries. This system is formed globally, and the local solutions are updated accordingly:

$$\begin{bmatrix} I & \tilde{E}_1 & & \\ \tilde{F}_2 & I & \ddots & \\ & \ddots & \ddots & \tilde{E}_{p-1} \\ & & \tilde{F}_p & I \end{bmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_p \end{pmatrix} = \begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \\ \tilde{c}_p \end{pmatrix}, \quad \text{where} \quad \begin{aligned} \tilde{E}_i &= [E_i^1, E_i^{n_i}]^T \in I\!R^2 \\ \tilde{F}_i &= [F_i^1, F_i^{n_i}]^T \in I\!R^2 \\ \tilde{x}_i &= [x_i^1, x_i^{n_i}]^T \in I\!R^2 \\ \tilde{c}_i &= [c_i^1, c_i^{n_i}]^T \in I\!R^2 \end{aligned}$$

Benefits:

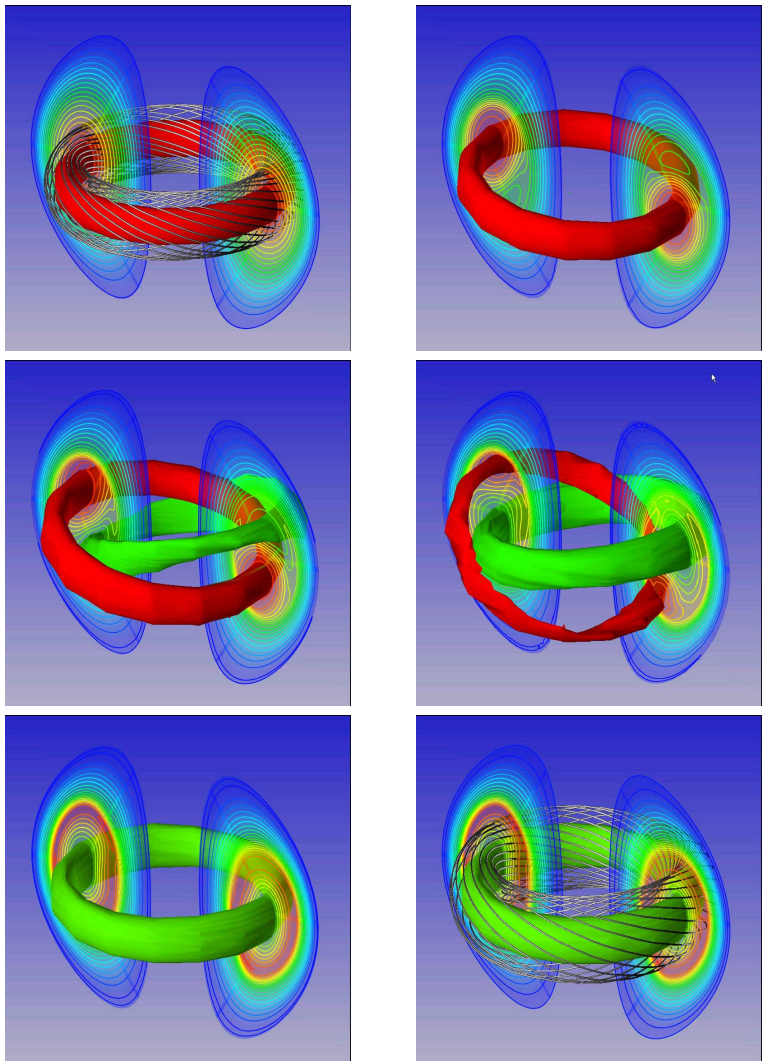
- Two sets of global communication: once to set up 1D MPI sub-communicators, once at each solve for global system.
- Non-iterative, i.e. 'true' solution is known at end of second phase.
- Easily generalizable to arbitrary boundary conditions.

[see Arbenz & Gander, ETH Zürich, Technical Report, 1994]

Magnetic Reconnection

Breaking and reconnecting of oppositely-directed magnetic field lines, converts magnetic energy to kinetic and thermal.

- Replaces hot plasma core with cool plasma, halting fusion processes
- ITER estimates require $S \approx 10^9$
- Spatial resolution scales as $S^{-1/2}$
- T_{recon} scales as $S^{1/2}$
- Simulations therefore require *large-scale*, *long-time* simulations (implicit)

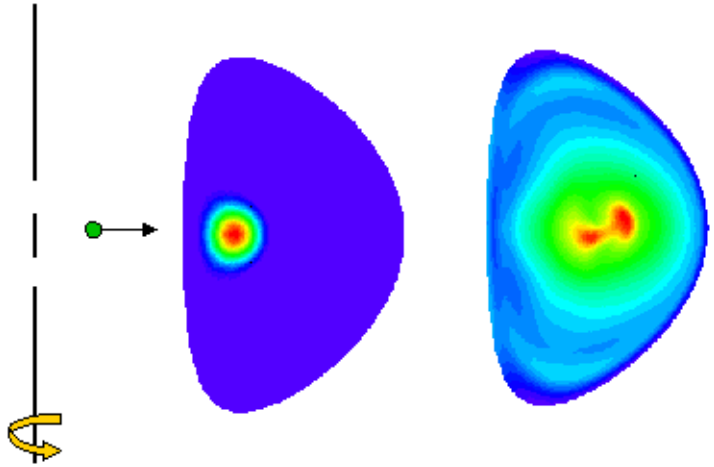


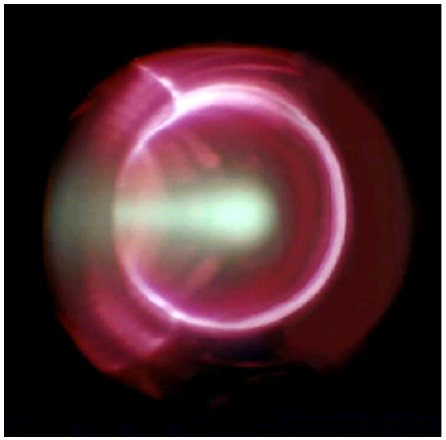
[see Brin et al., GEM magnetic reconnection challenge, *J. Geophys. Res.*, 106, 2001.]

Pellet Injection – Fueling the Reaction

Refuel the fusion reaction by shooting frozen hydrogen pellets into the plasma at high velocity (∼500 m/s)

- Ablation process well understood
- Mass deposition mostly MHD driven but not well understood
- Pellet motion $\mathcal{O}(10^4)$ slower than fastest waves, pellet size $\mathcal{O}(10^4)$ smaller than reactor
- Accurate modeling requires highly anisotropic heat conduction





[see Kuteev, *Nucl. Fusion*, 35, 1995; Samtaney et al., *Comp. Phys. Comm.*, 164, 2004]